### **Лабораторная работа №5**

**По МДК 02.03 «Математическое моделирование»**  
**Тема:** Задача о распределении средств между предприятиями

#### **Цель работы:**

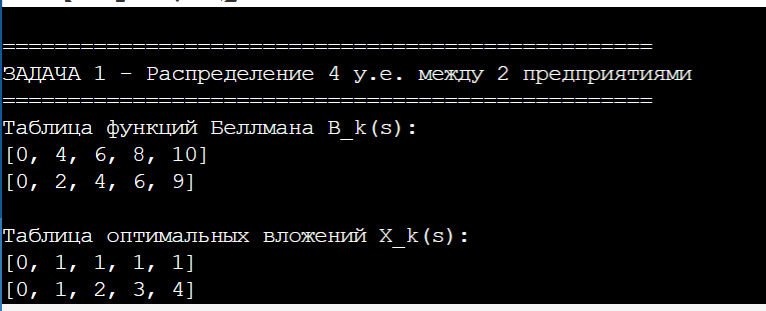
Научиться применять метод динамического программирования для решения оптимизационной задачи распределения ограниченного ресурса (капиталовложений) между несколькими предприятиями с целью максимизации совокупной прибыли.

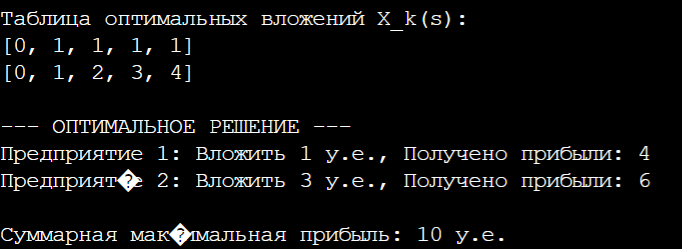
**Задача 1 (Простая):**  
Имеется 4 у.е. для распределения между 2 предприятиями.

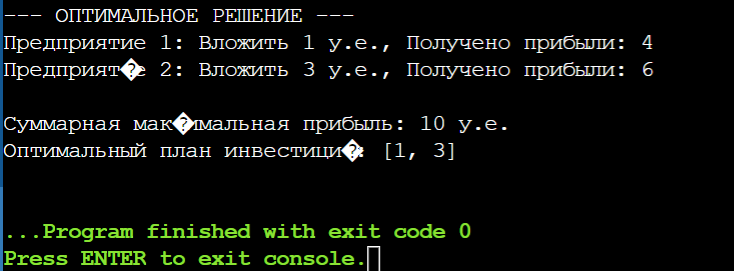
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Средства (x) | 0 | 1 | 2 | 3 | 4 |
| f1(x) | 0 | 4 | 6 | 7 | 8 |
| f2(x) | 0 | 2 | 4 | 6 | 9 |

**Вопрос:** Найдите оптимальный план распределения средств и максимальную суммарную прибыль.

Ответ:









print("=" \* 50)

print("ЗАДАЧА 1 - Распределение 4 у.е. между 2 предприятиями")

print("=" \* 50)

profit = [

[0, 4, 6, 7, 8],

[0, 2, 4, 6, 9]

]

n = len(profit)

S = 4

B\_table = [[0] \* (S + 1) for \_ in range(n)]

X\_table = [[0] \* (S + 1) for \_ in range(n)]

k = n - 1

for s in range(S + 1):

B\_table[k][s] = profit[k][s]

X\_table[k][s] = s

for k in range(n - 2, -1, -1):

for s in range(S + 1):

max\_profit = -1

best\_x = 0

for x in range(0, s + 1):

current\_profit = profit[k][x] + B\_table[k + 1][s - x]

if current\_profit > max\_profit:

max\_profit = current\_profit

best\_x = x

B\_table[k][s] = max\_profit

X\_table[k][s] = best\_x

print("Таблица функций Беллмана B\_k(s):")

for row in B\_table:

print(row)

print("\nТаблица оптимальных вложений X\_k(s):")

for row in X\_table:

print(row)

print("\n--- ОПТИМАЛЬНОЕ РЕШЕНИЕ ---")

remaining\_funds = S

total\_profit = 0

optimal\_investment = []

for k in range(n):

x\_opt = X\_table[k][remaining\_funds]

optimal\_investment.append(x\_opt)

profit\_k = profit[k][x\_opt]

total\_profit += profit\_k

print(f"Предприятие {k + 1}: Вложить {x\_opt} у.е., Получено прибыли: {profit\_k}")

remaining\_funds -= x\_opt

print(f"\nСуммарная максимальная прибыль: {total\_profit} у.е.")

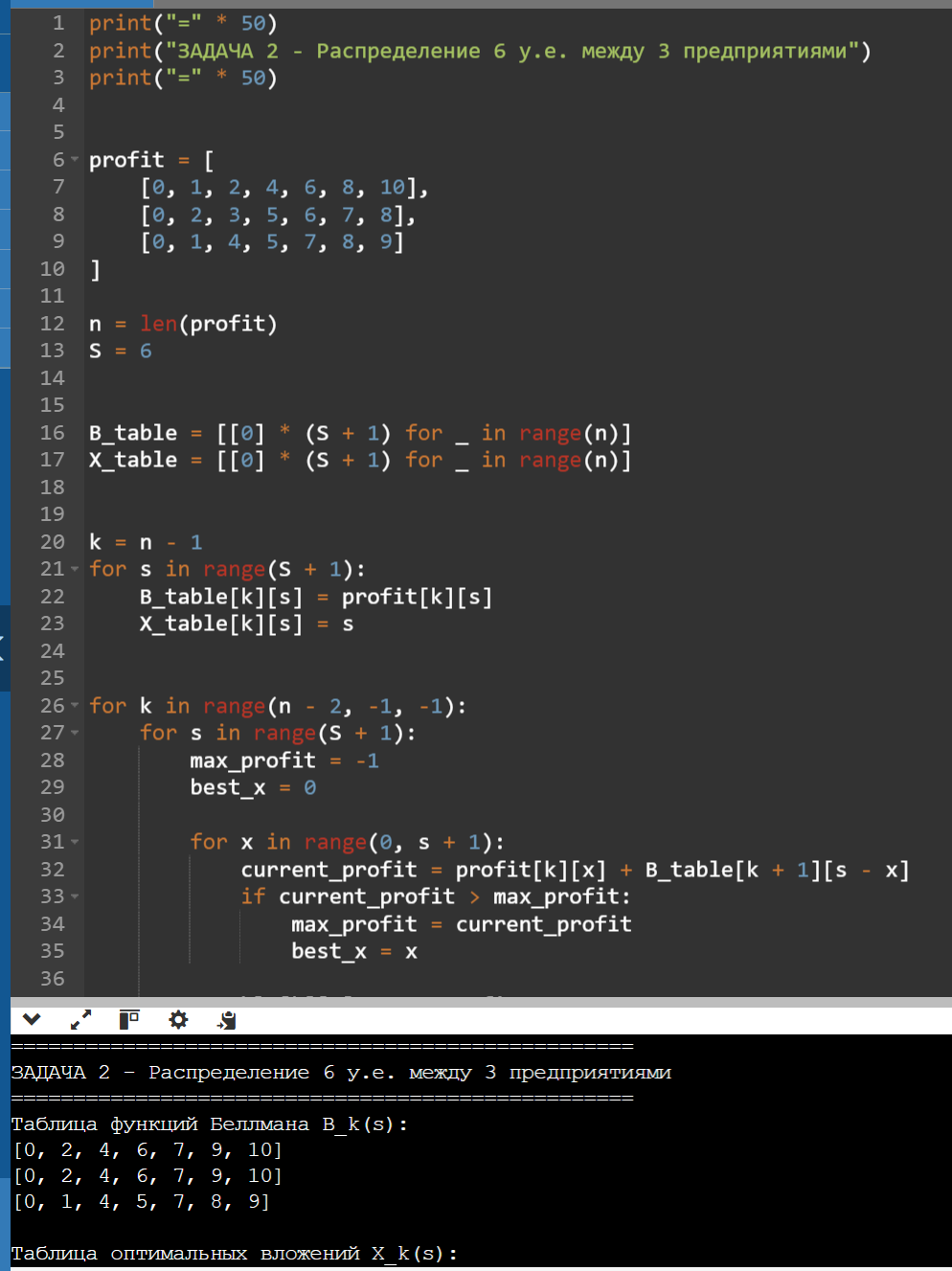
print(f"Оптимальный план инвестиций: {optimal\_investment}")

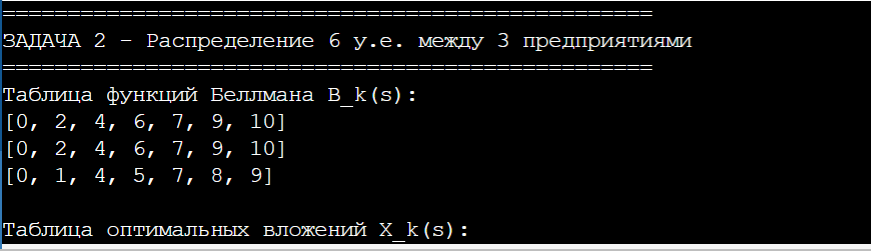
**Задача 2 (Средней сложности):**  
Требуется распределить 6 у.е. между 3 предприятиями.

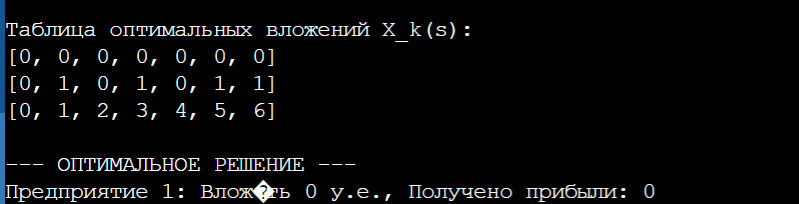
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Средства (x) | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| f1(x) | 0 | 1 | 2 | 4 | 6 | 8 | 10 |
| f2(x) | 0 | 2 | 3 | 5 | 6 | 7 | 8 |
| f3(x) | 0 | 1 | 4 | 5 | 7 | 8 | 9 |

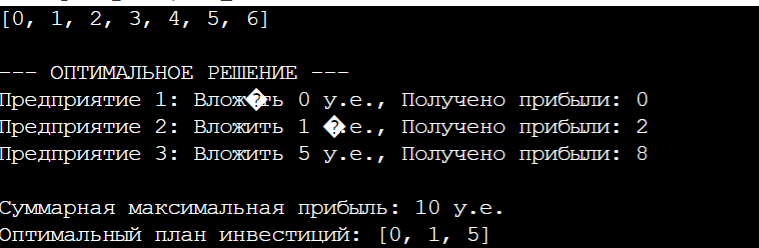
**Вопрос:** Каков оптимальный план инвестиций?

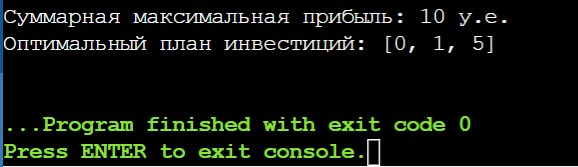
Ответ:











print("=" \* 50)

print("ЗАДАЧА 2 - Распределение 6 у.е. между 3 предприятиями")

print("=" \* 50)

profit = [

[0, 1, 2, 4, 6, 8, 10],

[0, 2, 3, 5, 6, 7, 8],

[0, 1, 4, 5, 7, 8, 9]

]

n = len(profit)

S = 6

B\_table = [[0] \* (S + 1) for \_ in range(n)]

X\_table = [[0] \* (S + 1) for \_ in range(n)]

k = n - 1

for s in range(S + 1):

B\_table[k][s] = profit[k][s]

X\_table[k][s] = s

for k in range(n - 2, -1, -1):

for s in range(S + 1):

max\_profit = -1

best\_x = 0

for x in range(0, s + 1):

current\_profit = profit[k][x] + B\_table[k + 1][s - x]

if current\_profit > max\_profit:

max\_profit = current\_profit

best\_x = x

B\_table[k][s] = max\_profit

X\_table[k][s] = best\_x

print("Таблица функций Беллмана B\_k(s):")

for row in B\_table:

print(row)

print("\nТаблица оптимальных вложений X\_k(s):")

for row in X\_table:

print(row)

print("\n--- ОПТИМАЛЬНОЕ РЕШЕНИЕ ---")

remaining\_funds = S

total\_profit = 0

optimal\_investment = []

for k in range(n):

x\_opt = X\_table[k][remaining\_funds]

optimal\_investment.append(x\_opt)

profit\_k = profit[k][x\_opt]

total\_profit += profit\_k

print(f"Предприятие {k + 1}: Вложить {x\_opt} у.е., Получено прибыли: {profit\_k}")

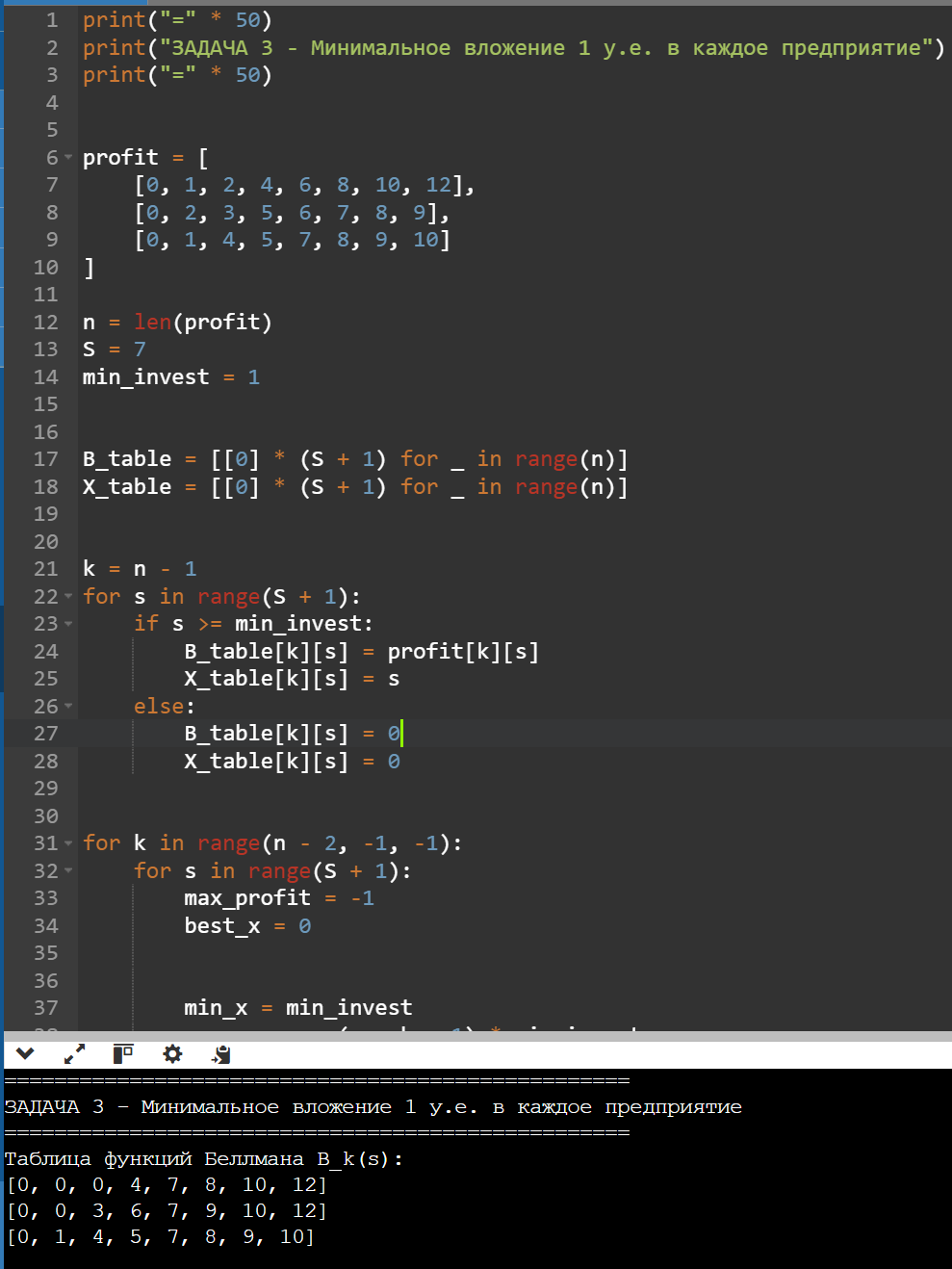
remaining\_funds -= x\_opt

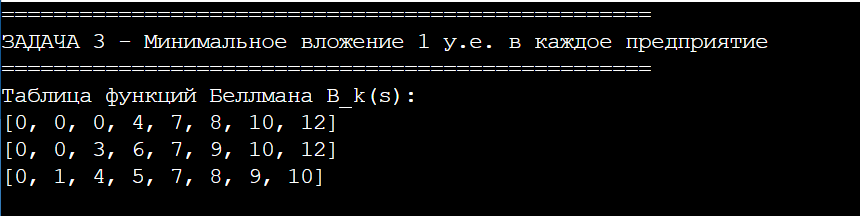
print(f"\nСуммарная максимальная прибыль: {total\_profit} у.е.")

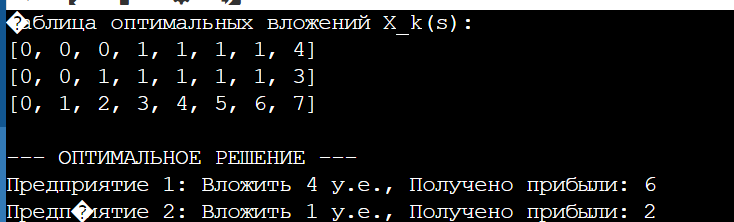
print(f"Оптимальный план инвестиций: {optimal\_investment}")

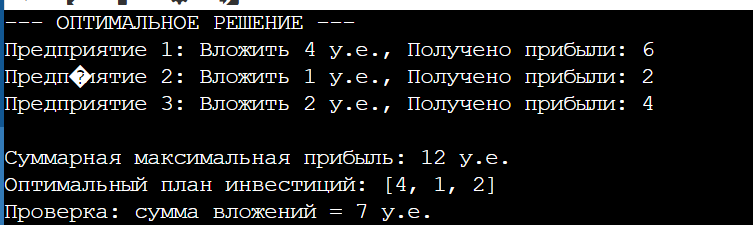
**Задача 3 (На подумать):**  
Как изменится алгоритм и код, если мы хотим в каждое предприятие вложить **хотя бы по 1 у.е.**? Попробуйте модифицировать программу для случая, когда начальная сумма S = 7, а предприятий 3 (из Задачи 2), и минимальное вложение в каждое — 1 у.е.

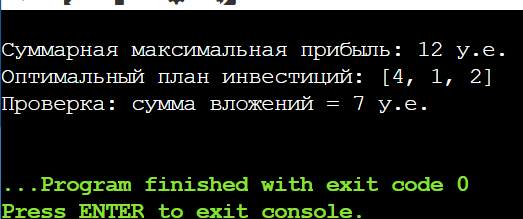
Ответ:











print("=" \* 50)

print("ЗАДАЧА 3 - Минимальное вложение 1 у.е. в каждое предприятие")

print("=" \* 50)

profit = [

[0, 1, 2, 4, 6, 8, 10, 12],

[0, 2, 3, 5, 6, 7, 8, 9],

[0, 1, 4, 5, 7, 8, 9, 10]

]

n = len(profit)

S = 7

min\_invest = 1

B\_table = [[0] \* (S + 1) for \_ in range(n)]

X\_table = [[0] \* (S + 1) for \_ in range(n)]

k = n - 1

for s in range(S + 1):

if s >= min\_invest:

B\_table[k][s] = profit[k][s]

X\_table[k][s] = s

else:

B\_table[k][s] = 0

X\_table[k][s] = 0

for k in range(n - 2, -1, -1):

for s in range(S + 1):

max\_profit = -1

best\_x = 0

min\_x = min\_invest

max\_x = s - (n - k - 1) \* min\_invest

for x in range(min\_x, max\_x + 1):

if x <= s and (s - x) >= (n - k - 1) \* min\_invest:

current\_profit = profit[k][x] + B\_table[k + 1][s - x]

if current\_profit > max\_profit:

max\_profit = current\_profit

best\_x = x

if max\_profit == -1:

B\_table[k][s] = 0

X\_table[k][s] = 0

else:

B\_table[k][s] = max\_profit

X\_table[k][s] = best\_x

print("Таблица функций Беллмана B\_k(s):")

for row in B\_table:

print(row)

print("\nТаблица оптимальных вложений X\_k(s):")

for row in X\_table:

print(row)

print("\n--- ОПТИМАЛЬНОЕ РЕШЕНИЕ ---")

remaining\_funds = S

total\_profit = 0

optimal\_investment = []

for k in range(n):

x\_opt = X\_table[k][remaining\_funds]

optimal\_investment.append(x\_opt)

profit\_k = profit[k][x\_opt]

total\_profit += profit\_k

print(f"Предприятие {k + 1}: Вложить {x\_opt} у.е., Получено прибыли: {profit\_k}")

remaining\_funds -= x\_opt

print(f"\nСуммарная максимальная прибыль: {total\_profit} у.е.")

print(f"Оптимальный план инвестиций: {optimal\_investment}")

print(f"Проверка: сумма вложений = {sum(optimal\_investment)} у.е.")